3D Textureless Object Detection and Tracking: An Edge-based Approach

Changhyun Choi and Henrik I. Christensen
Center for Robotics & Intelligent Machines
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
{cchoi,hic}@cc.gatech.edu

***Abstract*— This paper presents an approach to textureless object detection and tracking of the 3D pose. Our detection and tracking schemes are coherently integrated in a particle filtering framework on the special Euclidean group, SE(3), in which the visual tracking problem is tackled by maintaining multiple hypotheses of the object pose. For textureless object detection, an efficient chamfer matching is employed so that a set of coarse pose hypotheses is estimated from the matching between 2D edge templates of an object and a query image. Particles are then initialized from the coarse pose hypotheses by randomly drawing based on costs of the matching. To ensure the initialized particles are at or close to the global optimum, an annealing process is performed after the initialization. While a standard edge-based tracking is employed after the annealed initialization, we employ a refinement process to establish improved correspondences between projected edge points from the object model and edge points from an input image. Comparative results for several image sequences with clutter are shown to validate the effectiveness of our approach.**

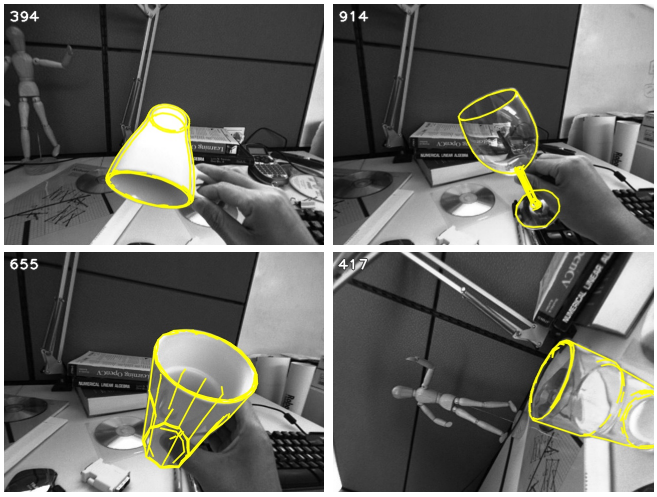

Fig. 1. **Example frames from our detection and tracking results.** Our approach combines detection and tracking for textureless objects in a particle filtering framework, and it employs edges as key visual information. It is capable of handling transparent objects, though it does not assume objects' transparency. Mean of particles is drawn in yellow wireframe on each image.

I. Introduction

In the last decade, object detection and recognition have significantly progressed based on keypoint features [1]. Since keypoints are invariant to geometric transformations and illumination changes, they have been widely used for matching similar images took from slightly different viewpoints [2]. Keypoint-based approaches are well suited for textured objects, but may not be effective for textureless objects because the features lacks repeatability and stability on textureless regions. Like keypoints, edges are also invariant to general geometric transformations and illumination changes [3], and they can be dependably detected for textureless objects.

In early computer vision research, an important problem was to find the best alignment between two edge maps. A set of edge templates of an object is known *a priori*, and the templates are searched in an edge map of query image. As a robust metric, the chamfer distance [4] was proposed, and there were several variants to enhance the cost functions by incorporating edge orientation [5], [6] and to reduce complexity by organizing templates in a hierarchical structure [7] or by employing integral images on linear representations of edges [8].

While these matching methods are expected to find exact shape matching, edges or contours have also been employed to solve object categorization problems [9], [10] in which the primary goal is to find a category over the intra-class variations. These efforts have shown promising performance on challenging image data. However, when a 3D geometric representation of an object is available and the goal is to find an exact object, the chamfer matching is generally preferred.

Visual tracking has also exploited edges [11] or contours [12]. Following the seminal work of Harris [11], various edge-based visual tracking systems [13], [14] have been proposed. One drawback of using edges is that they are not distinctive enough to provide effective discrimination. Since this disadvantage leads to failure in complex background or occlusions, there have been efforts to enhance the previous one by unifying interest points [15], [16] or considering multiple hypotheses on edge correspondences [17], [16]. But these efforts typically only considered a small number of hypotheses.

For consideration of multiple hypotheses in a more general sense particle filters have been proposed. After Isard and Blake [12] presented a particle filter method for a chal-

lenging 2D tracking problem, various particle filters have widely been proposed in 2D affine tracking with incremental measurement learning [18], [19] or 3D visual tracking [20], [21], [22].

II. Contributions

We propose an approach combining detection and tracking for textureless objects that is developed within a particle filtering framework. Especially, a particle filter on the *SE(3)* group is considered because it is geometrically meaningful and coordinate invariant, which means that noise distribution is independent of the choice of coordinates. Hence, overall tracking performance does not depend on the coordinates [23], [18].

Our key contributions are as follows:

- We employ an efficient chamfer matching to find a set of starting states. Most particle filtering approaches assume that an initial state is given or is searched from scratch with the simulated annealing [21]. Several have presented keypoint-based initialization [20], but keypoints are not usually applicable to textureless objects. Thus we present a 3D pose estimation from a chamfer matching [8] using a set of 2D edge templates.
- Although initial particles are assigned via the coarse pose hypotheses, they would be occasionally stuck in local optima right after the initialization. To ensure that initial states are at or close to the global optimum, we run a particle annealing method [24] right after the (re-)initialization.
- We refine edge correspondences between the projected model edges and the image edges via a RANSAC [25]. Most of the edge-based tracking approaches have used the nearest edges without performing refining process [13], [14], [11], except a few work [26], [22]. Considering the edge correspondences directly affect the measurement likelihood and thus entire tracking performance, we employ a RANSAC approach to ensure consistent edge data associations.

This paper is organized as follows. We introduce our particle filtering framework in Section III-A and III-B. The initialization scheme is then presented as the chamfer matching-based pose estimation, followed by the annealed particle filtering in Section III-C. After explaining the measurement refinement in Section III-D and particle optimization in Section III-E, discussion about symmetric objects and the re-initialization scheme are introduced in Section III-F and Section III-G, respectively. Finally, experimental results on various image sequences are shown in Section IV.

III. Particle Filter on the *SE(3)* Group

A. State Equations

The discrete system equation on the *SE(3)* group is acquired via the first-order exponential Euler discretization from the continuous general state equation [23]:

$$\mathbf{X}_t = \mathbf{X}_{t-1} \cdot \exp\left(\mathbf{A}(\mathbf{X}, t)\Delta t + \mathbf{dW}_t\sqrt{\Delta t}\right), \quad (1)$$

$$\mathbf{dW}_t = \sum_{i=1}^{6} \epsilon_{t,i}\mathbf{E}_i,\ \epsilon_t \sim \mathcal{N}(\mathbf{0}_{6\times 1}, \mathbf{\Sigma}_w)$$

where $\mathbf{X}_t \in SE(3)$ is the state at time t, $\exp : \mathfrak{se}(3) \mapsto SE(3)$ is the exponential map, $\mathbf{A} : SE(3) \mapsto \mathfrak{se}(3)$ is a possibly nonlinear map, $\mathbf{dW}_t$ represents the Wiener process noise on $\mathfrak{se}(3)$ with a covariance $\mathbf{\Sigma}_w \in \Re^{6\times 6}$, $\mathbf{E}_i$ are the i^{th} basis element of $\mathfrak{se}(3)$:

$$\mathbf{E}_1=\begin{pmatrix} 0&0&0&1\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}, \mathbf{E}_2=\begin{pmatrix} 0&0&0&0\\0&0&0&1\\0&0&0&0\\0&0&0&0\end{pmatrix}, \mathbf{E}_3=\begin{pmatrix} 0&0&0&0\\0&0&0&0\\0&0&0&1\\0&0&0&0\end{pmatrix},$$

$$\mathbf{E}_4=\begin{pmatrix} 0&0&0&0\\0&0&-1&0\\0&1&0&0\\0&0&0&0\end{pmatrix}, \mathbf{E}_5=\begin{pmatrix} 0&0&1&0\\0&0&0&0\\-1&0&0&0\\0&0&0&0\end{pmatrix}, \mathbf{E}_6=\begin{pmatrix} 0&-1&0&0\\1&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}. \quad (2)$$

The corresponding measurement equation is:

$$\mathbf{Z}_t = g(\mathbf{X}_t) + \mathbf{n}_t,\ \mathbf{n}_t \sim \mathcal{N}(\mathbf{0}_{\mathsf{N_Z}\times 1}, \mathbf{\Sigma}_n) \quad (3)$$

where $g : \mathbf{X}_t \mapsto \Re^{\mathsf{N_Z}}$ is a nonlinear measurement function and $\mathbf{n}_t$ is a Gaussian noise with a covariance $\mathbf{\Sigma}_n \in \Re^{\mathsf{N_Z}\times\mathsf{N_Z}}$.

A dynamic model for state evolution is essential since it has a significant impact on tracking performance. The first order auto-regressive (AR) state dynamics is a simple yet effective model as shown in [20], [18]. The term $\mathbf{A}(\mathbf{X}, t)$ in (1) determines the state dynamics. The first-order AR process on the Lie group can be modeled as

$$\mathbf{X}_t = \mathbf{X}_{t-1} \cdot \exp\left(\mathbf{A}_{t-1} + \mathbf{dW}_t\sqrt{\Delta t}\right), \quad (4)$$

$$\mathbf{A}_{t-1} = \lambda_{ar} \log(\mathbf{X}_{t-2}^{-1}\mathbf{X}_{t-1}) \quad (5)$$

where λ_{ar} is the AR process parameter and $\log : SE(3) \mapsto \mathfrak{se}(3)$ is the logarithmic map.

B. Particle Filter

In a particle filtering framework, the posterior density function $p(\mathbf{X}_t|\mathbf{Z}_{1:t})$ is represented as a set of weighted particles by

$$\mathcal{S}_t = \{(\mathbf{X}_t^{(1)}, \pi_t^{(1)}), \ldots, (\mathbf{X}_t^{(N)}, \pi_t^{(N)})\} \quad (6)$$

where the particles $\mathbf{X}_t^{(n)} \in SE(3)$ represent samples of the true state $\mathcal{X}_t$, the normalized weights $\pi_t^{(n)}$ are proportional to the likelihood function $p(\mathbf{Z}_t|\mathbf{X}_t^{(n)})$, and N is the number of particles. The current state $\mathcal{X}_t$ could be estimated by the weighted particle mean:

$$\mathcal{X}_t = \mathcal{E}[\mathcal{S}_t] = \sum_{n=1}^{N} \pi_t^{(n)}\mathbf{X}_t^{(n)}. \quad (7)$$

When we apply the mean, however, there is a problem where the average of $\mathbf{X}_t^{(n)}$ is not valid in the $SE(3)$. More specifically, let $\mathbf{R}_t^{(n)} \in SO(3)$ be the rotation part of the $\mathbf{X}_t^{(n)}$. Then the arithmetic mean $\overline{\mathbf{R}}_t = \frac{1}{N}\sum_{n=1}^{N}\mathbf{R}_t^{(n)}$ is not usually on the $SO(3)$ group. As an alternative, Moakher [27]

Algorithm 1: Particle Filtering on the *SE(3)* group

Data: $\mathcal{I} = \{\mathcal{I}_0, \mathcal{I}_1, \cdots, \mathcal{I}_I\}, \mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T\}$
Result: $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, \cdots, \mathcal{S}_I\}$
Params: $\mathbf{\Sigma}_w, \lambda_{ar}, \lambda_v, \lambda_e$

$t \leftarrow 0;\ init \leftarrow 1;\ \mathbf{A}_0 \leftarrow \mathbf{0}_{4\times4}$
while $\mathcal{I}_t \neq 0$ **do**
  if $init = 1$ **then**
    $\mathcal{S}_t \leftarrow \texttt{ChamferPose}(\mathcal{I}_t, \mathcal{T})$ ⟨2⟩
    $\mathcal{S}_t \leftarrow \texttt{ParticleAnnealing}(\mathcal{I}_t, \mathcal{S}_t)$ ⟨3⟩
    $init \leftarrow 0$
  else
    for $n \leftarrow 1$ **to** N **do**
      $\mathbf{X}_t^{*(n)} \leftarrow \texttt{Propagate}(\mathbf{X}_t^{(n)}, \mathbf{A}_{t-1}^{(n)}, \mathbf{\Sigma}_w)$ (4)
      $\mathbf{A}_t^{*(n)} \leftarrow \texttt{AR\_vel}(\mathbf{X}_t^{*(n)}, \mathbf{X}_{t-1}^{*(n)}, \lambda_{ar})$ (5)
      $\mathbf{Z}_t^{*(n)} \leftarrow \texttt{Measurement}(\mathbf{X}_t^{*(n)}, \mathcal{I}_t)$ (3)
      $\mathbf{Z}_t^{*(n)} \leftarrow \texttt{RANSAC}(\mathbf{Z}_t^{*(n)})$ ⟨4⟩
      $\pi_t^{*(n)} \leftarrow \texttt{Likelihood}(\mathbf{Z}_t^{*(n)}, \lambda_v, \lambda_e)$ (22)
      $\widehat{\mathbf{X}}_t^{*(n)} \leftarrow \texttt{IRLS}(\mathbf{X}_t^{*(n)}, \mathbf{Z}_t^{*(n)})$ (23)(24)
      $\widehat{\mathbf{Z}}_t^{*(n)} \leftarrow \texttt{Measurement}(\widehat{\mathbf{X}}_t^{*(n)}, \mathcal{I}_t)$ (3)
      $\widehat{\mathbf{Z}}_t^{*(n)} \leftarrow \texttt{RANSAC}(\widehat{\mathbf{Z}}_t^{*(n)})$ ⟨4⟩
      $\widehat{\pi}_t^{*(n)} \leftarrow \texttt{Likelihood}(\widehat{\mathbf{Z}}_t^{*(n)}, \lambda_v, \lambda_e)$ (22)
    $\widetilde{\mathcal{S}}_t^* \leftarrow \mathcal{S}_t^* \cup \widehat{\mathcal{S}}_t^*$
    for $n \leftarrow 1$ **to** $2N$ **do**
      $\widetilde{\pi}_t^{*(n)} \leftarrow \texttt{CorrectWeight}(\widetilde{\mathbf{X}}_t^{*(n)}, \widetilde{\pi}_t^{*(n)})$
    $\widetilde{\pi}_t^* \leftarrow \texttt{Normalize}(\widetilde{\pi}_t^*)$ (17)
    $\widehat{N_{eff}} \leftarrow \texttt{Neff}(\widetilde{\pi}_t^*)$ (30)
    if $\widehat{N_{eff}} \geq N_{thres}$ **then**
      $\mathcal{S}_t \leftarrow \texttt{Resampling}(\widetilde{\mathcal{S}}_t^*)$
    else
      $init \leftarrow 1$
  $t \leftarrow t + 1$

showed that a valid average of a set of rotations can be calculated by the orthogonal projection of $\overline{\mathbf{R}}_t$ as

$$\mathbf{R}_t = \begin{cases} \mathbf{V}\mathbf{U}^\mathsf{T} & \text{when } \det(\overline{\mathbf{R}}_t^\mathsf{T}) > 0 \\ \mathbf{V}\mathbf{H}\mathbf{U}^\mathsf{T} & \text{otherwise,} \end{cases} \tag{8}$$

where $\mathbf{U}$ and $\mathbf{V}$ are estimated via the singular value decomposition of $\overline{\mathbf{R}}_t^\mathsf{T}$ (*i.e.* $\overline{\mathbf{R}}_t^\mathsf{T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\mathsf{T}$) and $\mathbf{H} = \mathrm{diag}[1, 1, -1]$. Therefore, the valid arithmetic mean of the particles can be determined as

$$\mathcal{X}_t = \mathcal{E}_{SE(3)}[\mathcal{S}_t] = \begin{pmatrix} \mathbf{R}_t & \mathbf{T}_t \\ \mathbf{0}_{1\times3} & 1 \end{pmatrix} \tag{9}$$

where $\mathbf{T}_t = \frac{1}{N}\sum_{n=1}^{N} \mathbf{T}_t^{(n)}$ and $\mathbf{T}_t^{(n)} \in \Re^3$ is the translation part of $\mathbf{X}_t^{(n)}$.

The overall particle filtering algorithm is shown in Algorithm 1 where referred algorithms and equations are cited as $\langle\cdot\rangle$ and $(\cdot)$ in the comments area, respectively. It requires a sequence of images $\mathcal{I}$ and the edge templates $\mathcal{T}$ as an input and estimates the posterior density as a set of weighted particles $\mathcal{S}$ in each time t. Details of the algorithms and underlying models will be explained in subsequent sections.

C. Initialization

To initialize particles, coarse poses are estimated by employing an efficient chamfer matching [8] that provides sublinear time for the matching and shows fewer false positive rates via the piecewise smooth cost function. For this, a set

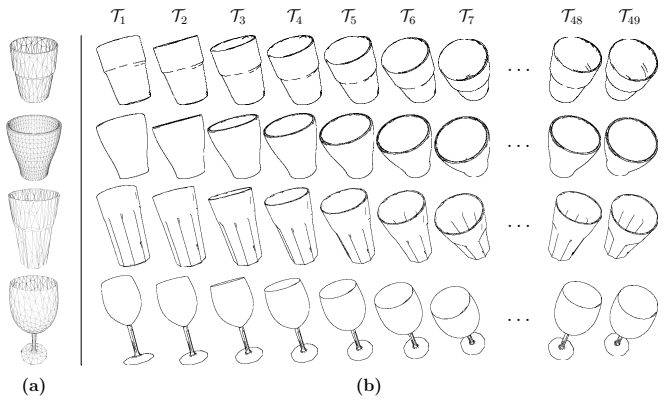

Fig. 2. **Polygonal mesh models and edge templates.** (a) We chose 4 IKEA objects so that replicating our experiments would be easier. From top to bottom, REKO glass, FARGRIK glass, POKAL glass, and SVALKA red wine glass. (b) Only visible edges were determined from the mesh models. To handle pose variations, the objects were rotated in $\mathbf{x}$ and $\mathbf{z}$ axes. These templates are used in the `ChamferPose` algorithm to estimate initial pose hypotheses.

of edge templates is obtained offline from polygonal mesh models as shown in Fig. 2.

1) Generating Edge Templates: We obtain edge templates $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T\}$ from the polygonal mesh models. To generate these templates, the projection matrix in OpenGL is set from the intrinsic camera parameters of the monocular camera which will be used in real experiments. The model is then rendered in OpenGL at a fixed depth Z_0. To identify visible edges, we use the face normal vectors from mesh models under an assumption that sharp edges would be more visible in real images. If the face normal vectors of two adjacent faces are close to perpendicular, the edge shared by the two faces is regarded as a sharp one. To determine if dull edges constitute boundaries of the objects, inner products of face normal vectors and the unit vector of $\mathbf{z}$-axis of camera coordinates are calculated. As appearances of the models change with respect to rotational variations, multiple templates are obtained as in Fig. 2 (b). To cover usual shape variations, the objects are rotated in $\mathbf{x}$ and $\mathbf{z}$ axes per 10° and 5°, respectively. Seven levels of rotations are sampled in each axis so that 49 templates are obtained per object.

2) Coarse Pose Estimation: With these templates, the chamfer matching is performed on an input image $\mathcal{I}_t$ across multi-scales. Among detection windows from the matching, we first consider windows under a threshold δ_{th}, then the non-maximum suppression is performed to have the lowest cost detection among the overlapped detection. As a result, we have a set of detections $\mathcal{D}$ for $m = 1, \ldots, M$:

$$\mathcal{D} = \{x^{(m)}, y^{(m)}, \delta^{(m)}, \mathcal{R}^{(m)}, \sigma^{(m)}\} \tag{10}$$

where $x^{(m)}$ and $y^{(m)}$ are the center location of the detected template in the input image, $\delta^{(m)}$ means the cost from the chamfer matching, $\mathcal{R}^{(m)} \in SO(3)$ is the corresponding rotation matrix saved in the template generation, $\sigma^{(m)}$ represents of the scale of the detected edge template, and M is the number of detections. The set of detections is sorted in order of increasing cost $\delta^{(m)}$.

Algorithm 2: `ChamferPose`$(\mathcal{I}, \mathcal{T})$

Data: $\mathcal{I}, \mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T\}$
Result: $\mathcal{S} = \{(\mathbf{X}^{(1)}, \pi^{(1)}), \cdots, (\mathbf{X}^{(N)}, \pi^{(N)})\}$
Params: $Z_0, u_0, v_0, f_x, f_y, \delta_{th}, \lambda_\delta$

$\mathcal{D} \leftarrow \{x^{(\varnothing)}, y^{(\varnothing)}, \delta^{(\varnothing)}, \mathcal{R}^{(\varnothing)}, \sigma^{(\varnothing)}\}$ (10)
for $t \leftarrow 1$ **to** T **do**
  for $\sigma \leftarrow \sigma_{min}$ **to** σ_{max} **do**
    $\{x', y', \delta', \mathcal{R}'\} \leftarrow$ `ChamferMatch`$(\mathcal{I}, \mathcal{T}_t, \sigma, \delta_{th})$ [8]
    $\mathcal{D}' \leftarrow \{x', y', \delta', \mathcal{R}'\} \cup \{\sigma\}$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}'$
`Sort`$(\mathcal{D})$
$M \leftarrow$ `length`$(\mathcal{D})$
for $m \leftarrow 1$ **to** M **do**
  $Z^{(m)} \leftarrow Z_0/\sigma^{(m)}$ (11)
  $X^{(m)} \leftarrow (x^{(m)} - u_0)Z^{(m)}/f_x$ (12)
  $Y^{(m)} \leftarrow (y^{(m)} - v_0)Z^{(m)}/f_y$ (13)
  $\mathcal{P}^{(m)} \leftarrow$ `CoarsePose`$(X^{(m)}, Y^{(m)}, Z^{(m)}, \mathcal{R}^{(m)})$ (14)
for $n \leftarrow 1$ **to** N **do**
  $\mathbf{X}^{*(n)} \leftarrow \mathcal{P}^{(n \bmod M+1)}$ (15)
  $\pi^{*(n)} \leftarrow \exp^{(-\lambda_\delta \delta^{(n \bmod M+1)})}$ (16)
$\pi^* \leftarrow$ `Normalize`(π^*) (17)
$\mathcal{S} \leftarrow$ `Resampling`$(\mathcal{S}^*)$

From $\mathcal{D}$, a set of coarse poses is estimated. As the edge templates do not cover the entire appearance variations, we can only approximate the current pose from the edge templates. For this approximation, two assumptions are considered. The first assumption is that although the center location $(x^{(m)}, y^{(m)})$ might be slightly far from the principal point (u_0, v_0), the rotation matrix can be adopted from the one at the principal point. Thus rotation of the object can be determined by $\mathbf{R}^{(m)}$. The second assumption is that the 3D center location of the object can be estimated via similar triangles in the perspective projection. Under this assumption, we can determine the z coordinate of the object $Z^{(m)}$ with respect to the camera by

$$Z^{(m)} = \frac{Z_0}{\sigma^{(m)}}. \tag{11}$$

Once $Z^{(m)}$ is determined, it is straightforward to calculate $X^{(m)}$ and $Y^{(m)}$ using similar triangles:

$$X^{(m)} = \frac{(x^{(m)} - u_0)}{f_x} Z^{(m)} \tag{12}$$

$$Y^{(m)} = \frac{(y^{(m)} - v_0)}{f_y} Z^{(m)} \tag{13}$$

where f_x and f_y are focal length in x and y directions of the camera, respectively. If we apply (11) to (12) and (13), we can represent the approximate pose hypothesis $\mathcal{P}^{(m)} \in SE(3)$ of the object with respect to the camera coordinates as follows

$$\mathcal{P}^{(m)} = \begin{pmatrix} & & \frac{(x^{(m)}-u_0)}{f_x}\frac{Z_0}{\sigma^{(m)}} \\ \mathcal{R}^{(m)} & & \frac{(y^{(m)}-v_0)}{f_y}\frac{Z_0}{\sigma^{(m)}} \\ & & \frac{Z_0}{\sigma^{(m)}} \\ \mathbf{0}_{1\times 3} & & 1 \end{pmatrix}. \tag{14}$$

Algorithm 3: `ParticleAnnealing`$(\mathcal{I}, \mathcal{S})$

Data: $\mathcal{I}, \mathcal{S} = \{(\mathbf{X}^{(1)}, \pi^{(1)}), \cdots, (\mathbf{X}^{(N)}, \pi^{(N)})\}$
Result: $\mathcal{S}_0 = \{(\mathbf{X}_0^{(1)}, \pi_0^{(1)}), \cdots, (\mathbf{X}_0^{(N)}, \pi_0^{(N)})\}$
Params: $\alpha = \{\alpha_0, \cdots, \alpha_L\}, \beta = \{\beta_0, \cdots, \beta_L\}, \mathbf{\Sigma}_{w,0}$

$\mathcal{S}_{L+1} \leftarrow \mathcal{S}$
for $l \leftarrow L$ **to** 0 **do**
  for $n \leftarrow 1$ **to** N **do**
    $\mathbf{X}_l^{*(n)} \leftarrow$ `Propagate`$(\mathbf{X}_{l+1}^{(n)}, \mathbf{\Sigma}_{w,l}, \alpha)$ (20)(21)
    $\mathbf{Z}^{*(n)} \leftarrow$ `Measurement`$(\mathbf{X}_l^{*(n)}, \mathcal{I})$ (3)
    $\acute{\mathbf{Z}}^{*(n)} \leftarrow$ `RANSAC`$(\mathbf{Z}^{*(n)})$
    $\pi_l^{*(n)} \leftarrow$ `Likelihood`$(\acute{\mathbf{Z}}^{*(n)}, \lambda_v, \lambda_e, \beta_l)$ (22)(19)
  $\pi_l^* \leftarrow$ `Normalize`(π_l^*) (17)
  $\mathcal{S}_l \leftarrow$ `Resampling`$(\mathcal{S}_l^*)$

After $\mathcal{P}^{(m)}$ is calculated for all M detection, the N particles and their weights are initialized as

$$\mathbf{X}^{*(n)} = \mathcal{P}^{(n \bmod M+1)} \tag{15}$$

$$\pi^{*(n)} = \exp^{(-\lambda_\delta \delta^{(n \bmod M+1)})} \tag{16}$$

where λ_δ is a parameter which controls the sensitivity for the costs. The weights are normalized via

$$\pi^{*(n)} = \frac{\pi^{*(n)}}{\sum_{i=1}^{N} \pi^{*(i)}}. \tag{17}$$

The particles are then randomly drawn with probability proportional to these weights, and we finally have a set of weighted particles $\mathcal{S}_t$ after the initialization. This pose estimation procedure is presented in Algorithm 2 where relevant paper and equations are cited as $[\cdot]$ and $(\cdot)$ in the comments area, respectively.

3) Annealed Particle Filtering: Although our particle filter starts with the most likely pose hypotheses, we cannot always guarantee that the filter converges to the global optimum. Since the sparse edge templates could not cover all possible ranges of pose variations, the errors come from this discrepancy might lead to local optima. Another limitation comes from the low precision of the chamfer matching. In cluttered backgrounds, the chamfer matching may return false positives which lead to poor initial states. Aside from these limitations, it is well known that even if a number of particles are employed, the particle filter might be stuck in local maxima.

To ensure that our particle filter starts near the global maximum, a simulated annealing [24] is performed after every initialization or re-initialization (Section III-G). The set of weighted particles in (6) is augmented with the annealing layer l as

$$\mathcal{S}_{t,l} = \{(\mathbf{X}_{t,l}^{(1)}, \pi_{t,l}^{(1)}), \ldots, (\mathbf{X}_{t,l}^{(N)}, \pi_{t,l}^{(N)})\}. \tag{18}$$

The annealing starts at layer $l = L$ where L is the number of annealing layers and the weights are determined by

$$\pi_{t,l}^{(n)} \propto p(\mathbf{Z}_t | \mathbf{X}_t^{(n)})^{\beta_l} \tag{19}$$

where β_l $(1 = \beta_0 > \beta_1 > \cdots > \beta_L)$ controls the rate of annealing at each layer. After normalization of the

weights, N particles are randomly drawn from $\mathcal{S}_{t,l}$ with the probability of their weights $\pi_{t,l}^{(n)}$. The particles of the next layer $\mathcal{S}_{t,l-1}$ are then propagated as

$$\mathbf{X}_{t,l-1}^{(n)} = \mathbf{X}_{t,l}^{(n)} \cdot \exp\left(\mathbf{dW}_{t,l}\sqrt{\Delta t}\right) \tag{20}$$

where $\mathbf{dW}_{t,l}$ is the Wiener process noise with covariance $\mathbf{\Sigma}_{w,l}$. This annealing process is iterated until it arrives at $\mathbf{X}_{t,0}^{(n)}$. In [24], the $\mathbf{\Sigma}_{w,l}$ was defined by

$$\mathbf{\Sigma}_{w,l} = \mathbf{\Sigma}_{w,0}(\alpha_L \alpha_{L-1} \dots \alpha_l) \tag{21}$$

where α_l represents the particle survival rate which is equivalent to $\widehat{N_{eff}}/N$ in (30). They argued that $\alpha_0 = \alpha_1 = \dots = \alpha_L = 0.5$ provide sufficient results. In the parameter β_l, one can determine to adjust an initial rate of α_{init} to α_l using a gradient descent method [24]. As a simple alternative, we empirically found $\beta_l = (0.5)^l$ shows good performance as well. The annealing algorithm is shown in Algorithm 3.

D. Edge-based Measurement Likelihood

In edge-based tracking, a set of visible edges from a 3D polygonal mesh model is projected according to a current pose hypothesis. Then a set of points is sampled along the visible edges per a fixed distance. The sampled points are then matched to the nearest edge pixels from the image by 1D perpendicular search [20], [14]. Once these matches are determined, the measurement likelihood is defined by the number of matched sample points p_m, the number of visible sample points p_v which pass a self-occlusion test, and the arithmetic average distances $\bar{e}$ between the matched sample points and the edge pixels as in [20]:

$$p(\mathbf{Z}_t|\mathbf{X}_t^{(n)}) \propto \exp^{(-\lambda_v \frac{(p_v - p_m)}{p_v})} \exp^{(-\lambda_e \bar{\mathbf{e}})} \tag{22}$$

where λ_v and λ_e control the sensitivity for each term. Unfortunately, this nearest neighbor matching often results in false matches due to background clutter, shadow, or non-Lambertian reflectance. These false matches give wrong measurement likelihood, and thus the false correspondences result in a bad state hypothesis. Some efforts tried to enhance these matches through maintaining multiple low-level edge clusters [22] or applying a RANSAC on each 2D line segments [26]. One drawback of both is the possibility of inconsistent refinement because edge or line segments are individually corrected.

For consistent refinement, we perform a RANSAC on 3D sampled points $\mathbf{P}$ and their corresponding 2D closest edge points $\mathbf{p}$. Our approach consistently discard outliers by estimating the best 3D pose containing the largest number of inliers $\hat{\mathcal{H}}$. The refining process is shown in Algorithm 4 where m is the minimum number of points to find a hypothesis $\tilde{\mathbf{X}}$, $\mathbf{K}$ is the 3×3 intrinsic camera matrix, and the `Projection` means the general perspective projection.

E. Optimization using IRLS

Local optimization on particles is preferred when we expect better accuracy with relatively a small number of particles. We minimize the error $\boldsymbol{e}$ by performing Iterative Re-weighted Least Squares (IRLS) [14]. From IRLS, the optimized particle $\widehat{\mathbf{X}}_t^{*(n)}$ is calculated as

$$\widehat{\mathbf{X}}_t^{*(n)} = \mathbf{X}_t^{*(n)} \cdot \exp\left(\sum_{i=1}^{6} \mu_i \mathbf{E}_i\right) \tag{23}$$

$$\boldsymbol{\mu} = (\mathbf{J}^\mathsf{T}\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^\mathsf{T}\mathbf{W}\boldsymbol{e} \tag{24}$$

where $\boldsymbol{\mu} \in \mathfrak{R}^6$ is the motion velocity that minimizes the error vector $\boldsymbol{e} \in \mathfrak{R}^{\mathsf{N_z}}$, $\mathbf{J} \in \mathfrak{R}^{\mathsf{N_z} \times 6}$ is a Jacobian matrix of $\boldsymbol{e}$ with respect to $\boldsymbol{\mu}$ obtained by computing partial derivatives at the current pose, and $\mathbf{W} \in \mathfrak{R}^{\mathsf{N_z} \times \mathsf{N_z}}$ is a weighted diagonal matrix. Detailed formulation can be found in [14].

After IRLS optimization, we have slightly different samples $\widehat{\mathbf{X}}_t^*$ from $\mathbf{X}_t^*$. Since the new samples were not sampled from the prior distribution $p(\mathbf{X}_t|\mathbf{Z}_{1:t-1})$, they are required to be corrected according to the importance sampling theory [12]. This correction can be done by applying the correction factor $f_t(\mathbf{X}_t^{*(n)})/g_t(\mathbf{X}_t^{*(n)})$ as in [28], [22]:

$$\pi_t^{*(n)} \propto \frac{f_t(\mathbf{X}_t^{*(n)})}{g_t(\mathbf{X}_t^{*(n)})} p(\mathbf{Z}_t|\mathbf{X}_t^{*(n)}) \tag{25}$$

where $f_t(\mathbf{X})$ is the approximated prior distribution as a mixture of Gaussians, $g_t(\mathbf{X})$ is also the approximated distribution in which the prior samples are combined with the optimized samples as

$$f_t(\mathbf{X}) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{N}((\log(\mathbf{X}_t^{*(n)}))^\vee, \mathbf{\Sigma}_f)(\mathbf{X}), \tag{26}$$

$$\hat{f}_t(\mathbf{X}) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{N}((\log(\widehat{\mathbf{X}}_t^{*(n)}))^\vee, \mathbf{\Sigma}_f)(\mathbf{X}), \tag{27}$$

$$g_t(\mathbf{X}) = \frac{1}{2}\left(f_t(\mathbf{X}) + \hat{f}_t(\mathbf{X})\right) \tag{28}$$

where the mapping $^\vee : \mathfrak{se}(3) \mapsto \mathfrak{R}^6$ is defined as $(\sum_{i=1}^{6} x_i \mathbf{E}_i)^\vee = (x_1, x_2, \cdots, x_6)^\mathsf{T}$, and $\mathbf{\Sigma}_f \in \mathfrak{R}^{6\times 6}$ is a covariance.

Algorithm 4: `RANSAC`$(\mathbf{X}, \mathbf{Z})$

Data: $\mathbf{X}, \mathbf{Z} = \{\mathbf{p}, \mathbf{P}\}$
Result: $\hat{\mathcal{H}}$
Params: $i_{max}, m, \mathbf{K}, \epsilon_{th}, \rho$

$i \leftarrow 0;\ \hat{n} \leftarrow 0;\ \kappa \leftarrow \infty$
$\hat{\mathcal{H}} \leftarrow \{\phi\};\ \mathcal{H} \leftarrow \{\phi\};\ nop \leftarrow$ `length`$(\mathbf{p})$
while $i < \kappa$ **and** $i < i_{max}$ **do**
  $\tilde{\mathbf{Z}} \leftarrow$ `RandomSample`$(\mathbf{Z}, m)$
  $\tilde{\mathbf{X}} \leftarrow$ `IRLS`$(\mathbf{X}, \tilde{\mathbf{Z}})$ (23)(24)
  $\acute{p} \leftarrow$ `Projection`$(\mathbf{K}, \tilde{\mathbf{X}}, P)$
  $\mathcal{H} = \{h \mid \|\mathbf{p}^{(h)} - \acute{\mathbf{p}}^{(h)}\|_2 < \epsilon_{th}\}$
  $n \leftarrow$ `length`$(\mathcal{H})$
  if $n > \hat{n}$ **then**
    $\hat{n} \leftarrow n;\ \hat{\mathcal{H}} \leftarrow \mathcal{H}$
    $\kappa \leftarrow \log(1-\rho)/\log(1-(\hat{n}/nop)^m)$
  $i \leftarrow i + 1$

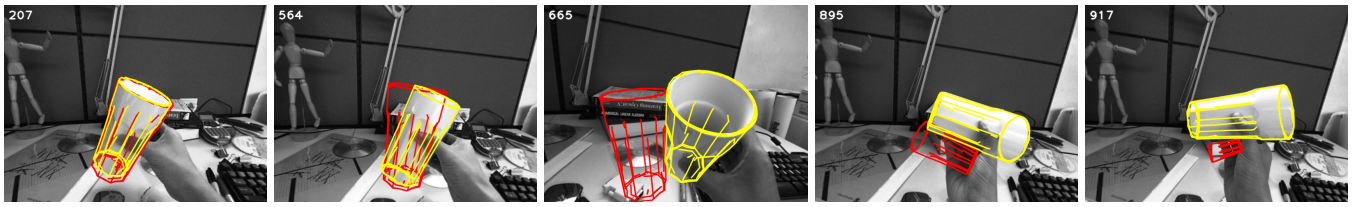

Fig. 3. **Tracking results showing effectiveness of considering multiple hypotheses.** Results with 100 particles (yellow wireframe) and 1 particle (red wireframe) are shown in the sequence of the POKAL glass. Note that the yellow wireframe is well localized by calculating the mean of multiple hypotheses, while the red wireframe is drifted during entire tracking. The frame number is shown in the top left corner of each image.

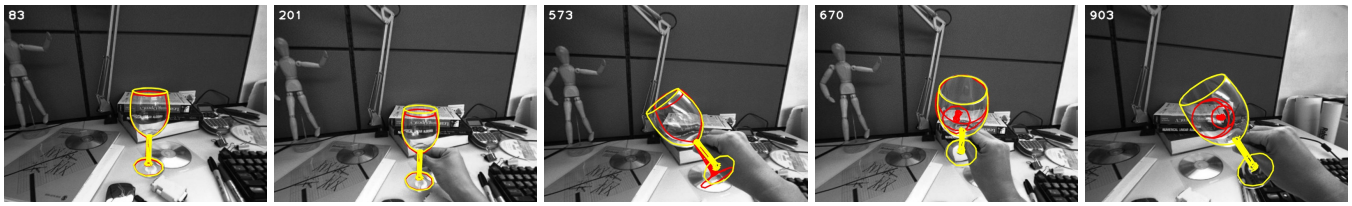

Fig. 4. **Tracking results showing effectiveness of performing the RANSAC.** Results with (yellow wireframe) and without (red wireframe) the refinement are shown in the wine glass sequence. While the yellow wireframe well follows the wine glass, the red wireframe is severely miss aligned.

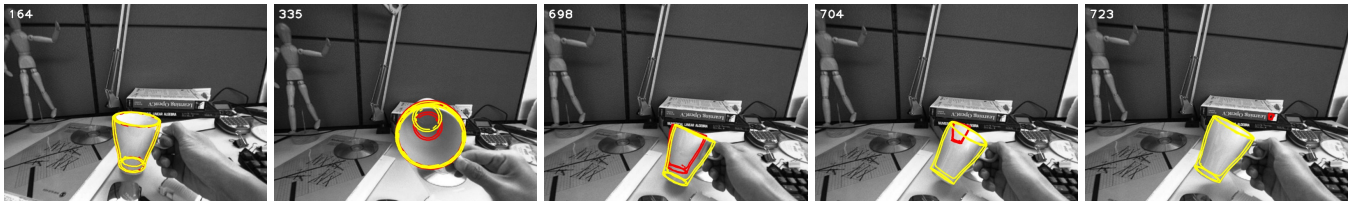

Fig. 5. **Tracking results showing effectiveness of suppressing the rotating motion about the axis of symmetry.** Results with (yellow wireframe) and without (red wireframe) the suppression are shown in the sequence of the FARGRIK glass. Although they use the same number of particles and the parameters, the red wireframe starts to drift before the frame number 698 due to larger search space.

F. Symmetric Objects

Some of our objects (Fig. 2) are symmetrical so that rotation about the axis of symmetry, $\mathbf{y}$-axis in our objects, cannot be uniquely determined. It is problematic when our particle filter searches the 6D pose space because it may result in a ridge posterior distribution. Thus, it is more efficient to search a 5D pose space instead of the full 6D space. This can be easily modified through our Lie group formulation. Recall that $\mathfrak{se}(3)$ has 6 basis elements as shown in (2), and exponentiating the term of the fifth basis $\mathbf{E}_5$ results in the rotation about $\mathbf{y}$-axis in $SE(3)$:

$$\exp(\gamma \mathbf{E}_5) = \begin{pmatrix} \cos\gamma & 0 & \sin\gamma & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\gamma & 0 & \cos\gamma & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{29}$$

Therefore, it is possible to suppress rotating motion about the axis of symmetry by setting 0 in the fifth coefficient for $\mathbf{E}_5$ corresponding to the term $\mathbf{A}(\mathbf{X}, t)$ and $\mathbf{dW}_t$ in (1), $\mathbf{dW}_{t,l}$ in (20), and $\boldsymbol{\mu}$ in (24).

G. Re-initialization

During visual tracking, it is quite common that the object goes out of sight or is occluded by other objects. In these cases, the tracker is required to re-initialize by itself. In [29], the effective particle size $\widehat{N_{eff}}$ has been introduced as a suitable measure of degeneracy:

$$\widehat{N_{eff}} = \frac{1}{\sum_{i=1}^{N} (\pi^{(i)})^2}. \tag{30}$$

As shown in [20], it can be used as a measure to do re-initialization. When the number of effective particles is below a fixed threshold N_{thres}, the re-initialization procedure is performed.

IV. Experimental Results

In this section, we validate our proposed solution using a number of comparative experiments. REKO, SVALKA, POKAL glasses were chosen from the KIT ObjectModels Web Database[1] in which more than 100 object models of household items are provided in 3D polygonal meshes and stereo images, and FARGRIK glass model obtained from Google 3D warehouse[2]. We only use the provided mesh models in our experiments which are shown in Fig. 2 (a). From these models, we prepare 49 edge templates per object offline (Fig. 2 (b)). These templates are used in the chamfer matching to initialize particles. To obtain test image sequences, a calibrated monocular camera was placed around the target objects, and the camera was moved so that the resulted sequences of images shows significant variation in translation, rotation, and velocity.

To validate our particle filtering approach, we first executed our system on the sequence of the POKAL glass with 1 and 100 particles. For fair comparison, we set the same parameters except the number of particles. In Fig. 3, results of the system using 1 and 100 particles are depicted

[1] http://wwwiaim.ira.uka.de/ObjectModels/
[2] http://sketchup.google.com/3dwarehouse/

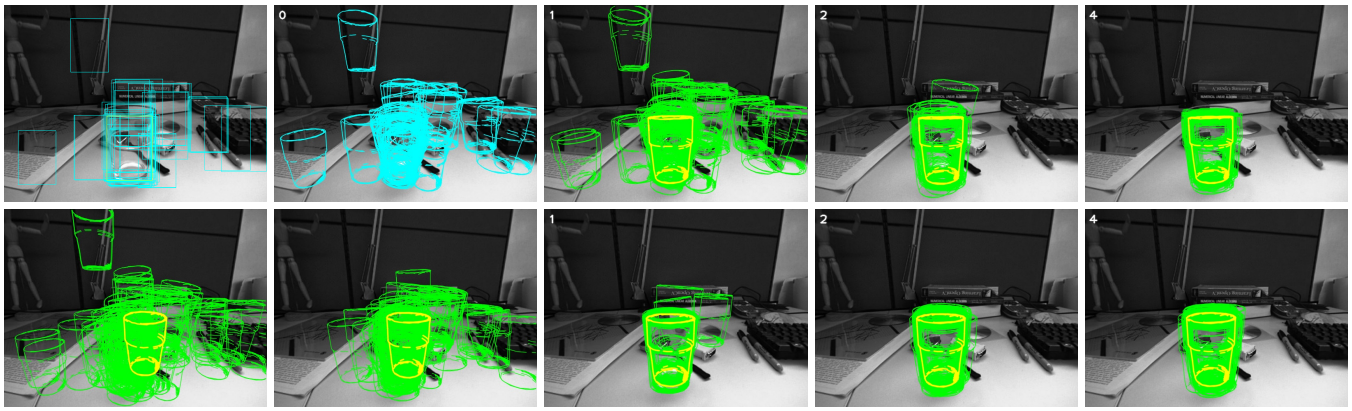

Fig. 6. **Initialization and annealed particle filtering.** The top-left image shows the chamfer matching results which are depicted in cyan bounding boxes, except that the lowest cost window (*i.e.* best result) is drawn in the yellow box. The next image shows initial states in cyan wireframes determined by the `ChamferPose` algorithm. The upper and lower rows of the center to right columns present results without and with the annealing, respectively. Intermediate annealing results are shown in the bottom-left two images (annealing layer l is 4 and 2 from total $L = 5$ layers). The particle filter without annealing is frequently stuck in local optima, and thus it could not recover to the global optimum, while our annealed particle filter can converge.

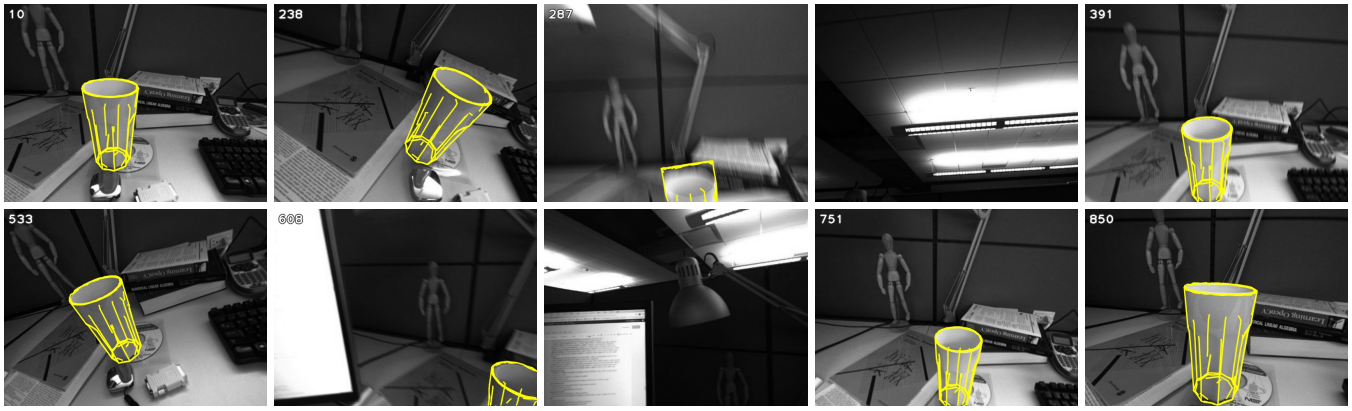

Fig. 7. **Tracking results showing the re-initialization capability.** Based on the value of $\widehat{N_{eff}}$, our system can re-initialize when the object goes out of the field of view.

in red and yellow wireframes, respectively. While the red wireframes are suffered from drifting, the yellow ones are well fitted to the object. Since the particle filter considers multiple hypotheses, it is not stuck in local optima.

We also evaluated the effectiveness of the RANSAC by executing our system with and without the refinement. Again, for fair comparison we used the same parameters. From the results in Fig. 4, we can verify that the RANSAC procedure enhances edge correspondences. Hence our approach shows more stable tracking than the one having no RANSAC.

To verify the effectiveness of suppressing the rotating motion discussed in Section III-F, the proposed approach was executed with and without the suppression. For fair comparison, the suppression was only altered. The tracking results are presented in Fig. 5. The tracking difference is possibly due to different search spaces. With the same number of particles ($N = 100$), the suppressed version only searches for the global optimum in the 5D space, while the version without the suppression fails to find the optimum in the 6D space.

We prove the effectiveness of annealed particle filtering by turning on and off the annealing stage after the initialization. Again the experiment was executed with the same parameters except the annealing. The comparison of the two tracking results are presented in Fig. 6. It is clear that employing the annealing process helps the tracker to start from the global optimum.

As monitoring the effective number of particles $\widehat{N_{eff}}$, the proposed system can re-initialize by itself when it is required. To verify this capability, we tested on a challenging image sequence in which the object is often disappeared because of the camera motion (Fig. 7). When these cases are occurred, $\widehat{N_{eff}}$ falls significantly. Thus our system re-initializes the tracking and successfully recovers from the failure cases.

V. Conclusions

We presented a particle filtering approach using edge features for the textureless object detection and tracking. Our approach started with possible pose hypotheses via the chamfer matching followed by the coarse pose estimation. The initial poses were further refined through the annealed particle filtering to ensure they are close to the global maximum. In addition, to handle false edges from non-Lambertian reflectance and clutter we employed the RANSAC refinement process which gave improved edge correspondences. The proposed approach was qualitatively validated in various experiments.

VI. ACKNOWLEDGMENTS

This work has in part been sponsored by the Boeing Corporation. The support is gratefully acknowledged.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[2] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *ICCV*, vol. 2, 2005, pp. 1458–1465 Vol. 2.

[3] J. Canny, "A computational approach to edge detection," *PAMI*, pp. 679–698, 1986.

[4] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching," in *IJCAI*, 1977, pp. 659–663.

[5] C. Olson and D. Huttenlocher, "Automatic target recognition by matching oriented edge pixels," *IEEE Transactions on Image Processing*, vol. 6, no. 1, pp. 103–113, 1997.

[6] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *CVPR*, 2010.

[7] D. M. Gavrila, "A Bayesian, exemplar-based approach to hierarchical shape matching," *PAMI*, pp. 1408–1421, 2007.

[8] M. Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *CVPR*, 2010, pp. 1696–1703.

[9] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid, "Groups of adjacent contour segments for object detection," *PAMI*, vol. 30, no. 1, pp. 36–51, 2008.

[10] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *PAMI*, vol. 30, no. 7, pp. 1270–1281, 2008.

[11] C. Harris, *Tracking with Rigid Objects*. MIT Press, 1992.

[12] M. Isard and A. Blake, "Condensation–conditional density propagation for visual tracking," *IJCV*, vol. 29, no. 1, pp. 5–28, 1998.

[13] A. I. Comport, E. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *IROS*, vol. 1, 2004.

[14] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *PAMI*, vol. 24, no. 7, pp. 932–946, 2002.

[15] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *ICCV*, vol. 2, 2005.

[16] L. Vacchetti, V. Lepetit, and P. Fua, "Combining edge and texture information for real-time accurate 3D camera tracking," in *ISMAR*, 2004, pp. 48–56.

[17] C. Kemp and T. Drummond, "Dynamic measurement clustering to aid real time tracking," in *ICCV*, 2005, pp. 1500–1507.

[18] J. Kwon and F. C. Park, "Visual tracking via particle filtering on the affine group," *IJRR*, vol. 29, no. 2-3, pp. 198–217, 2010.

[19] D. A. Ross, J. Lim, R. S. Lin, and M. H. Yang, "Incremental learning for robust visual tracking," *IJCV*, vol. 77, no. 1, pp. 125–141, 2008.

[20] C. Choi and H. I. Christensen, "Robust 3D visual tracking using particle filtering on the SE(3) group," in *ICRA*, 2011.

[21] G. Klein and D. Murray, "Full-3D edge tracking with a particle filter," *BMVC*, 2006.

[22] C. Teulière, E. Marchand, and L. Eck, "Using multiple hypothesis in model-based tracking," in *ICRA*, 2010.

[23] J. Kwon, M. Choi, F. C. Park, and C. Chun, "Particle filtering on the Euclidean group: framework and applications," *Robotica*, vol. 25, no. 06, pp. 725–737, 2007.

[24] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *CVPR*, vol. 2, 2000, pp. 126–133 vol.2.

[25] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[26] M. Armstrong and A. Zisserman, "Robust object tracking," in *ACCV*, vol. 1, 1995, pp. 58–61.

[27] M. Moakher, "Means and averaging in the group of rotations," *SIAM Journal on Matrix Analysis and Applications*, vol. 24, no. 1, pp. 1–16, 2003.

[28] M. Bray, E. Koller-Meier, and L. V. Gool, "Smart particle filtering for 3D hand tracking," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, 2004, pp. 675–680.

[29] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and computing*, vol. 10, no. 3, pp. 197–208, 2000.